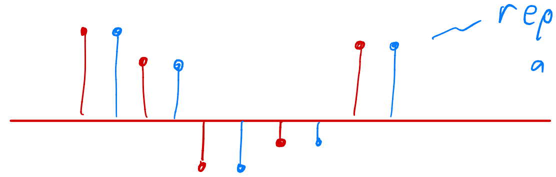
1-D Rate Conversion

- Decimation - lose information
  - Reduce the sampling rate of a discrete-time signal.
  - Low sampling rate reduces storage and computation requirements.
- Interpolation - redundant representation
  - Increase the sampling rate of a discrete-time signal.
  - Higher sampling rate preserves fidelity.
  - Like zooming in with 2 fingers on a phone/tablet

Nyquist band-limited

sample at $f \rightarrow$ no frequencies expected at $\frac{f}{2}$



replication by a factor of 2

CMF is strictly positive (corresponding primary is imaginary)

has some short wavelength filter

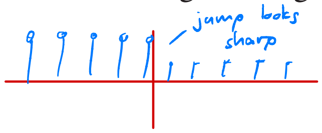
G	R
B	G

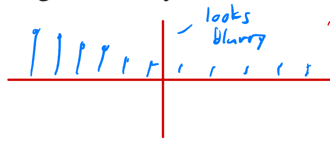
has some long wavelength filter

each cell is an individual pixel on a sensor and can measure independently

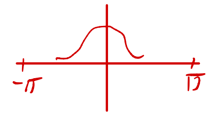
green (Y) has largest spatial frequency, most related to luminance

$x(\lambda)$, $y(\lambda)$, $z(\lambda)$ curves are important to get right or certain colors may be inaccurately misrepresented

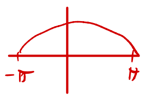




wastes resolution



desirable



1-D Periodic Subsampling

↳ a bad thing

- Time domain subsampling of $x(n)$ with period D

$$y(n) = x(Dn)$$

$$x(n) \longrightarrow \boxed{\downarrow D} \longrightarrow y(n)$$

- Frequency domain representation

$$Y(e^{j\omega}) = \frac{1}{D} \sum_{k=0}^{D-1} X\left(e^{j(\omega - 2\pi k)/D}\right)$$

where does the 2π go?

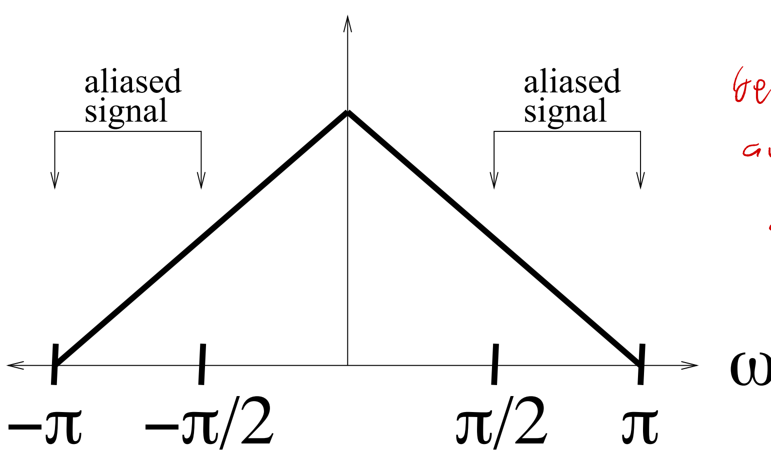
- Problem: Frequencies above π/D will alias.

like removing every other pixel

better to average groups of 4 or something similar

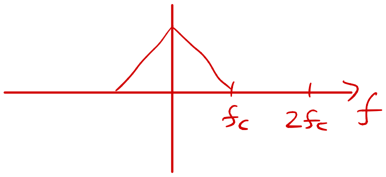
like applying a rect in time

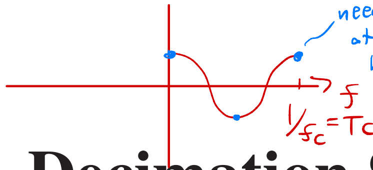
- Example when $D = 2$



- Solution: Remove frequencies above π/D.

Nyquist





need to sample at twice the highest frequency

$x_n = e^{j\pi n} = (-1)^n$ } max frequency in discrete time (2π)

Decimation System

$$x(n) \longrightarrow \boxed{H(e^{j\omega})} \xrightarrow{g(n)} \boxed{\downarrow D} \longrightarrow y(n)$$

$= s(m) = g(mD)$

$Y(e^{j\omega}) = \frac{1}{D} \sum_{-\infty}^{\infty} G\left(\frac{\omega - 2\pi k}{2\pi D}\right)$

↳ required so you don't alias

- Apply the filter $H(e^{j\omega})$ to remove high frequencies

  $\text{sinc}(n) = \delta(n)$

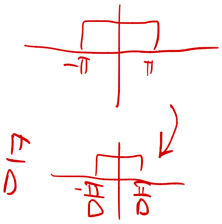
  – For $|\omega| < \pi$

  $G(e^{j\omega}) = X(e^{j\omega}) H(e^{j\omega})$

$$H(e^{j\omega}) = \text{rect}\left(\frac{D\omega}{2\pi}\right)$$

$\frac{D\omega}{2\pi} < \frac{1}{2}$

$\omega < \frac{2\pi}{2D} = \frac{\pi}{D}$



  – For all ω

$$\begin{aligned} H(e^{j\omega}) &= \sum_{k=-\infty}^{\infty} \text{rect}\left(D\frac{\omega - k2\pi}{2\pi}\right) \\ &= \text{prect}_{2\pi/D}(\omega) \end{aligned}$$

tied to sampling equations on cheat sheet

  – Impulse response

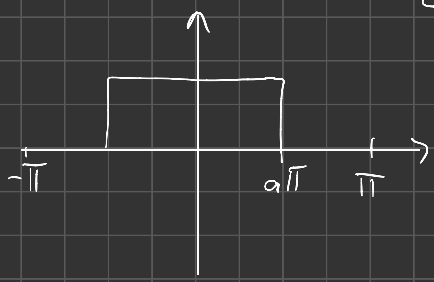
$$h(n) = \frac{1}{D}\text{sinc}(n/D)$$

- Frequency domain representation

$$Y(e^{j\omega}) = \frac{1}{D}\sum_{k=0}^{D-1} H\left(e^{j(\omega-2\pi k)/D}\right) X\left(e^{j(\omega-2\pi k)/D}\right)$$

where D is eliminating every D samples

$0 < a < 1$

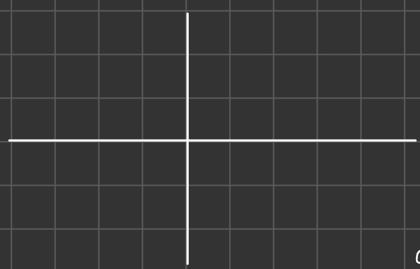


$f(t) = \mathrm{sinc}(at)$

$x_n = a\,\mathrm{sinc}(an)$ $\omega = 2\pi f$

$F(f) = \frac{1}{a}\,\mathrm{rect}\left(\frac{\omega}{2\pi a}\right)$

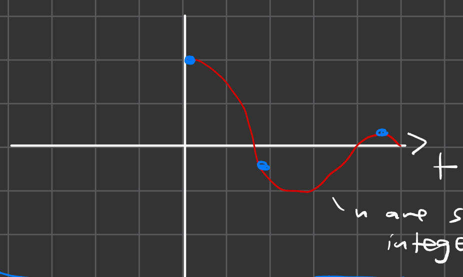
$X(e^{j\omega})$

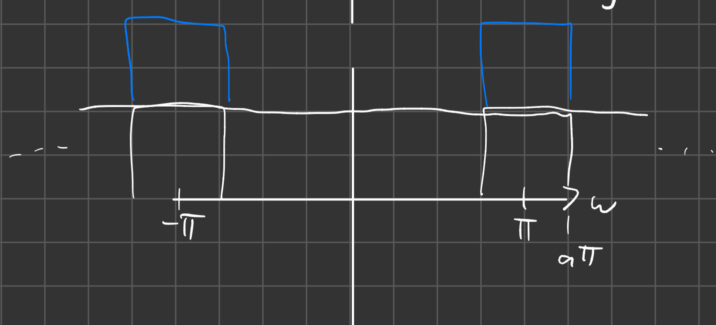


$a > 1$

$h_n = a\,\mathrm{sinc}(an)$ — as a increases you compress h_n, $H(e^{j\omega})$ stretches ⊛

$H(e^{j\omega}) = \mathrm{rect}\left(\frac{\omega}{2\pi a}\right)$



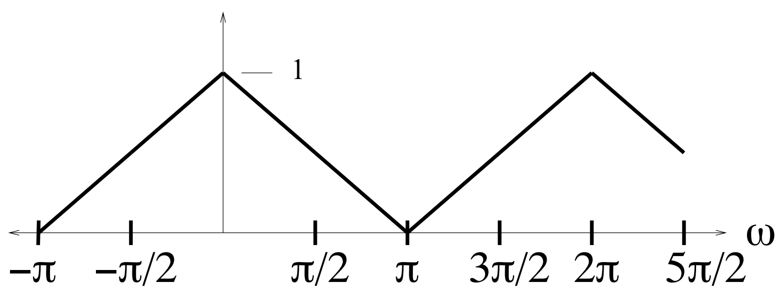


Graphical View of Decimation for $D = 2$

D=3

- Spectral content of signal

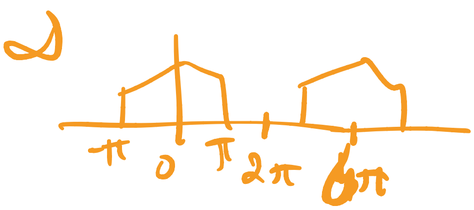
$$X\left(e^{j\omega}\right)$$

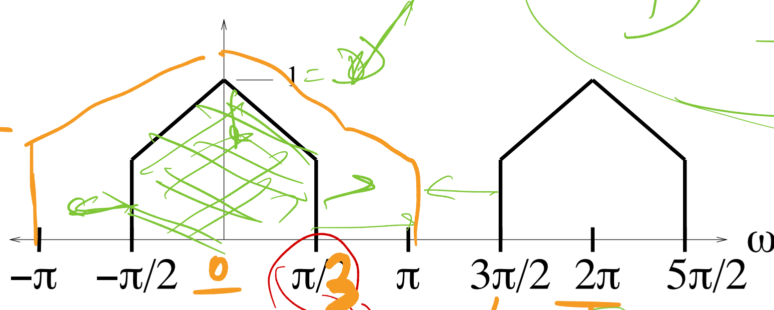


- Spectral content of filtered signal

$$H\left(e^{j\omega}\right) X\left(e^{j\omega}\right)$$

D = 1





$2\pi - \frac{\pi}{3} = \frac{5\pi}{3}$ → 6π

→ 5π

D = 2

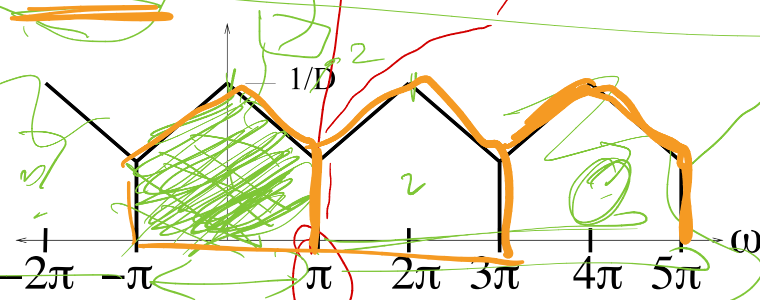
how π maps to decimation fraction

π − 5π

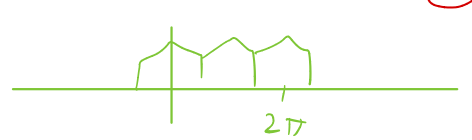
- Spectral content of decimated signal

$$Y\left(e^{j\omega}\right) = \frac{1}{D}\sum_{k=0}^{D-1} H\left(e^{j(\omega-2\pi k)/D}\right) X\left(e^{j(\omega-2\pi k)/D}\right)$$

2πD-periodic



D=3 →



Decimation for Images

- Extension to decimation of images is direct
- Apply 2-D Filter

$$f(i,j) = h(i,j) * x(i,j)$$

- Subsample result

$$y(i,j) = f(Di, Dj)$$

- Ideal choice of filter is

$$h(m,n) = \frac{1}{D^2}\text{sinc}(m/D)\text{sinc}(n/D)$$

computationally expensive

- Problems:
  - Filter has infinite extent.
  - Filter is not strictly positive.

Alternative Filters for Image Decimation

- Direct subsampling

$$h(m,n) = \delta(m,n)$$

  - Advantages/Disadvantages:
    - Low computation
    - Excessive aliasing

- Block averaging

FT is sinc-like

$$\begin{aligned} h(m,n) &= \big(\delta(m,n) + \delta(m+1,n) \\ &\quad + \delta(m,n+1) + \delta(m+1,n+1)\big)/4 \end{aligned}$$

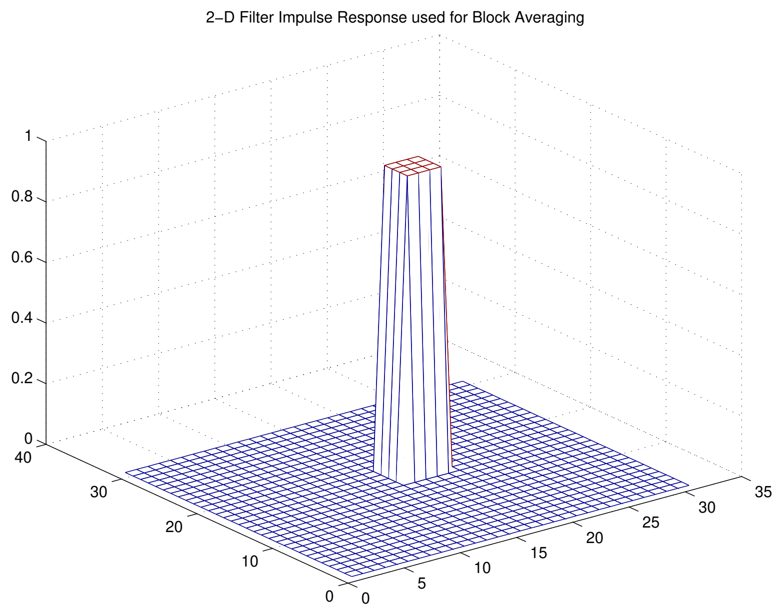
  - Advantages/Disadvantages:
    - Low computation
    - Some aliasing

- Sinc function

$$h(m,n) = \frac{1}{D^2}\text{sinc}(m/D, n/D)$$

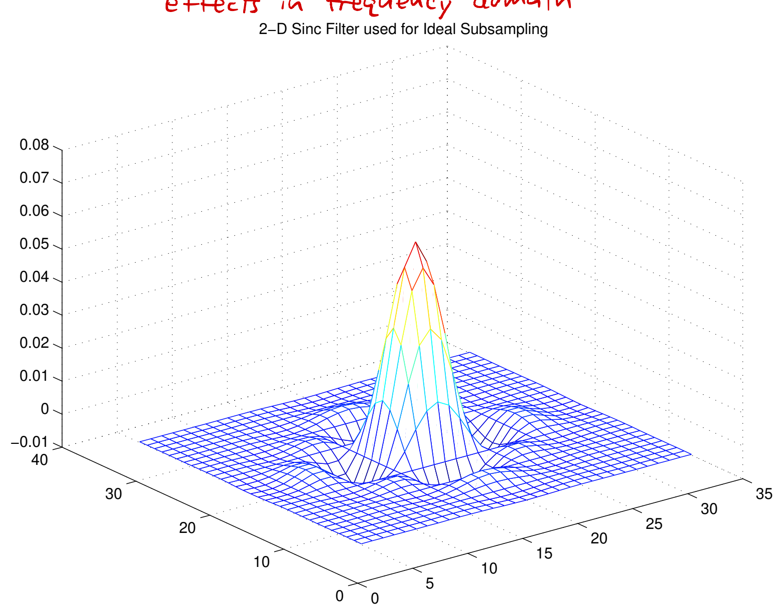
  - Advantages/Disadvantages:
    - Optimal, if signal is band limited...
    - High computation

Decimation Filters



Block averaging filter

↳ leads to some softening/low-pass effects in frequency domain



Sinc filter

Original Image

- Full resolution

Image Decimation by 4 using Subsampling

- Severe aliasing

Image Decimation by 8 using Subsampling

- More severe aliasing

Image Decimation by 4 using Block Averaging

- Sharp, but with some aliasing

Image Decimation by 4 using Sinc Filter

- Theoretically optimal, but not necessarily the best visual quality

1-D Up-Sampling

- Up-sampling by L

$$y(n) = \begin{cases} x(n/L) & \text{if } n = KL \text{ for some } K \\ 0 & \text{otherwise} \end{cases}$$

or the alternative form

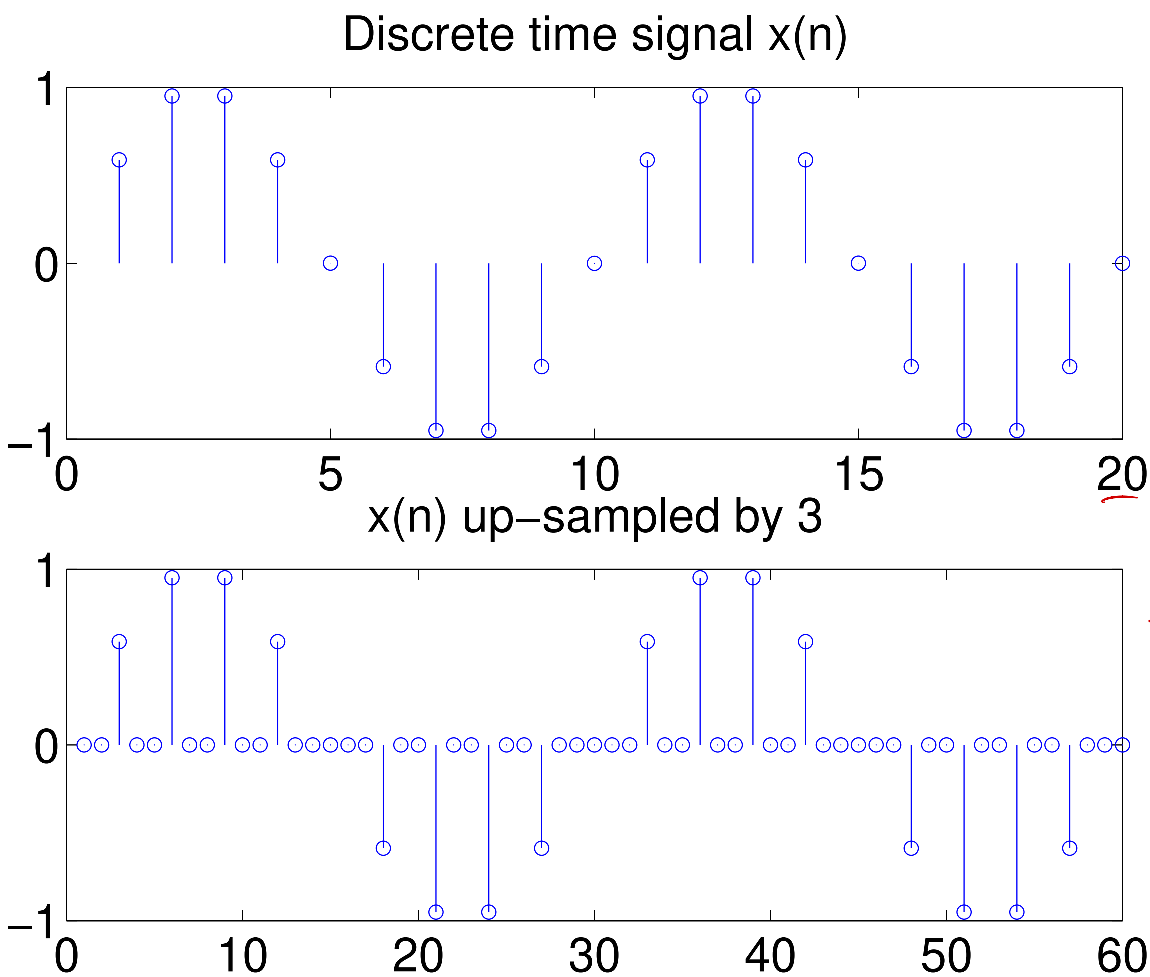
$$y(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n - mL)$$

"on" when $n = mL \rightarrow m = \frac{n}{L}$

$\frac{1}{2}, \frac{2}{2}, \frac{3}{2}$, etc

has more info (up-sampled)

- Example for $L = 3$



being stretched in time

Up-Sampling in the Frequency Domain

- Up-sampling by L

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n - mL)$$

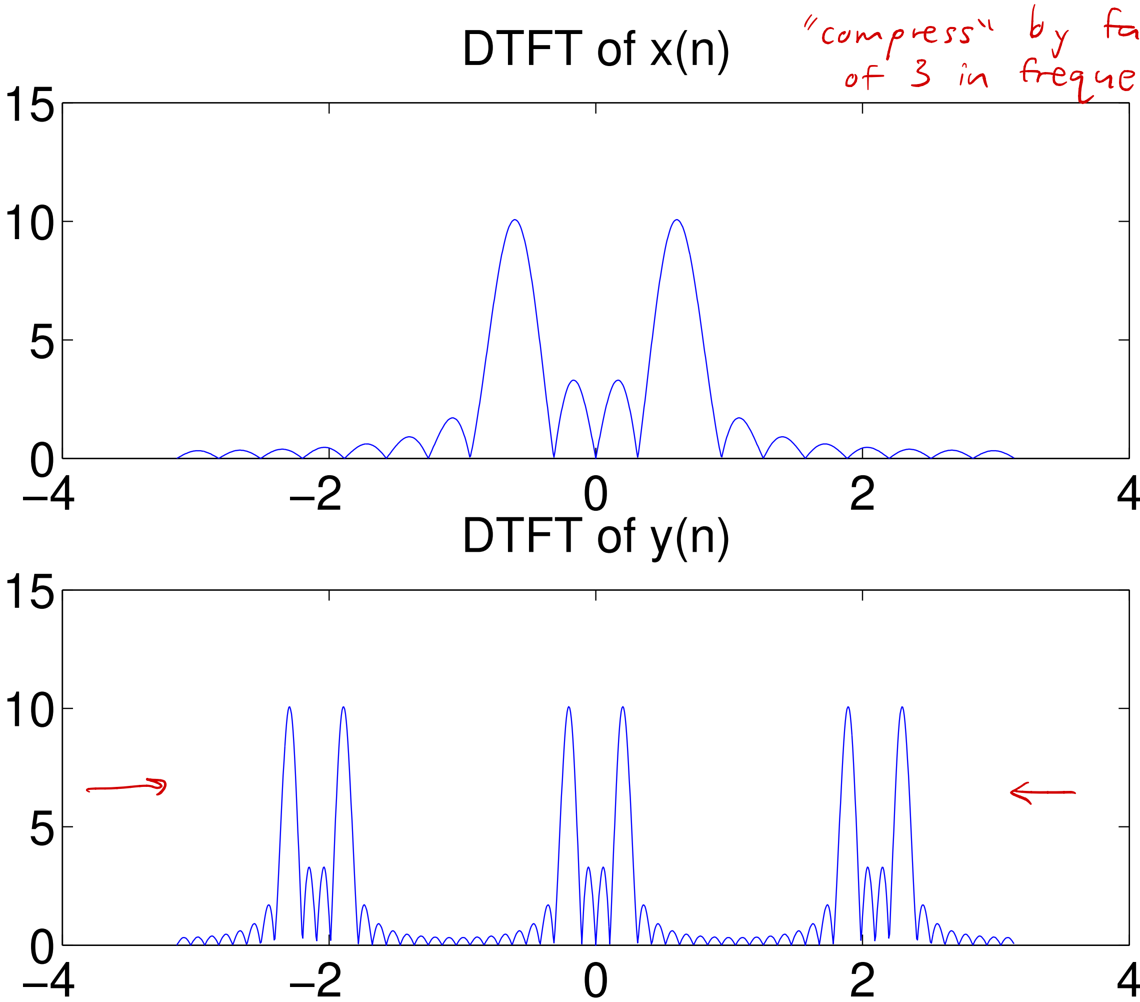
- In the frequency domain

say $\omega = \pi$: $Y(e^{j\pi}) = X(e^{j3\pi})$

$$Y(e^{j\omega}) = X(e^{j\omega L})$$

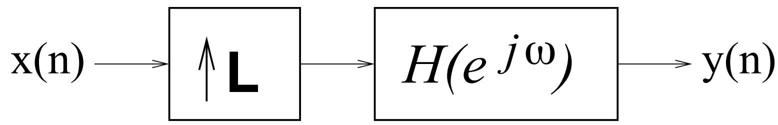
- Example for $L = 3$

"stretch" by factor of 3 in time, "compress" by factor of 3 in frequency ✱



sum of impulses responses is the DC gain which we choose to be 1

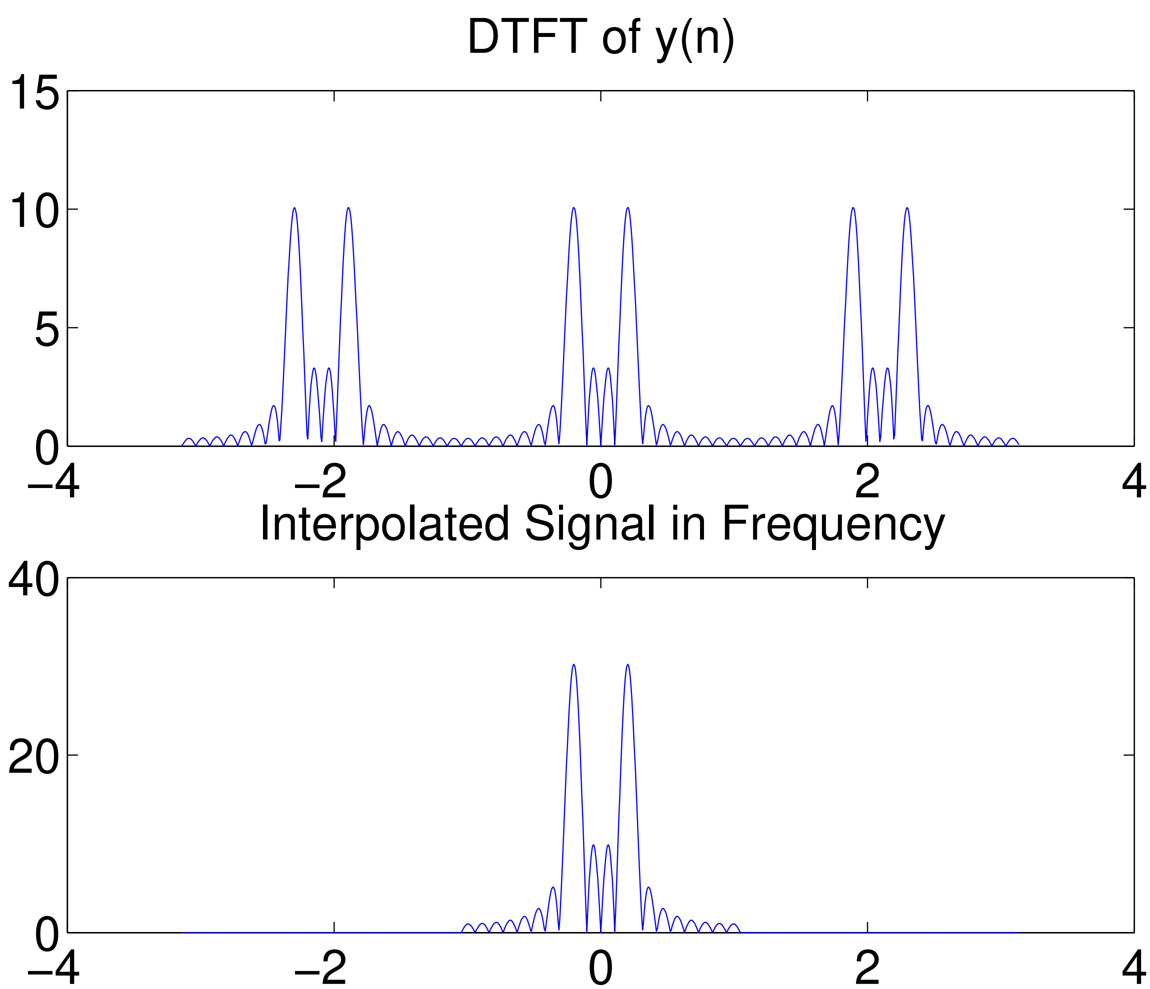
Interpolation in the Frequency Domain



- Interpolating filter has the form

$$Y(e^{j\omega}) = H(e^{j\omega})\,X(e^{j\omega L})$$

- Example for $L = 3$



Interpolating Filter

$$x(n) \longrightarrow \boxed{\uparrow \mathbf{L}} \longrightarrow \boxed{H(e^{j\omega})} \longrightarrow y(n)$$

- In the frequency domain

LPF in frequency

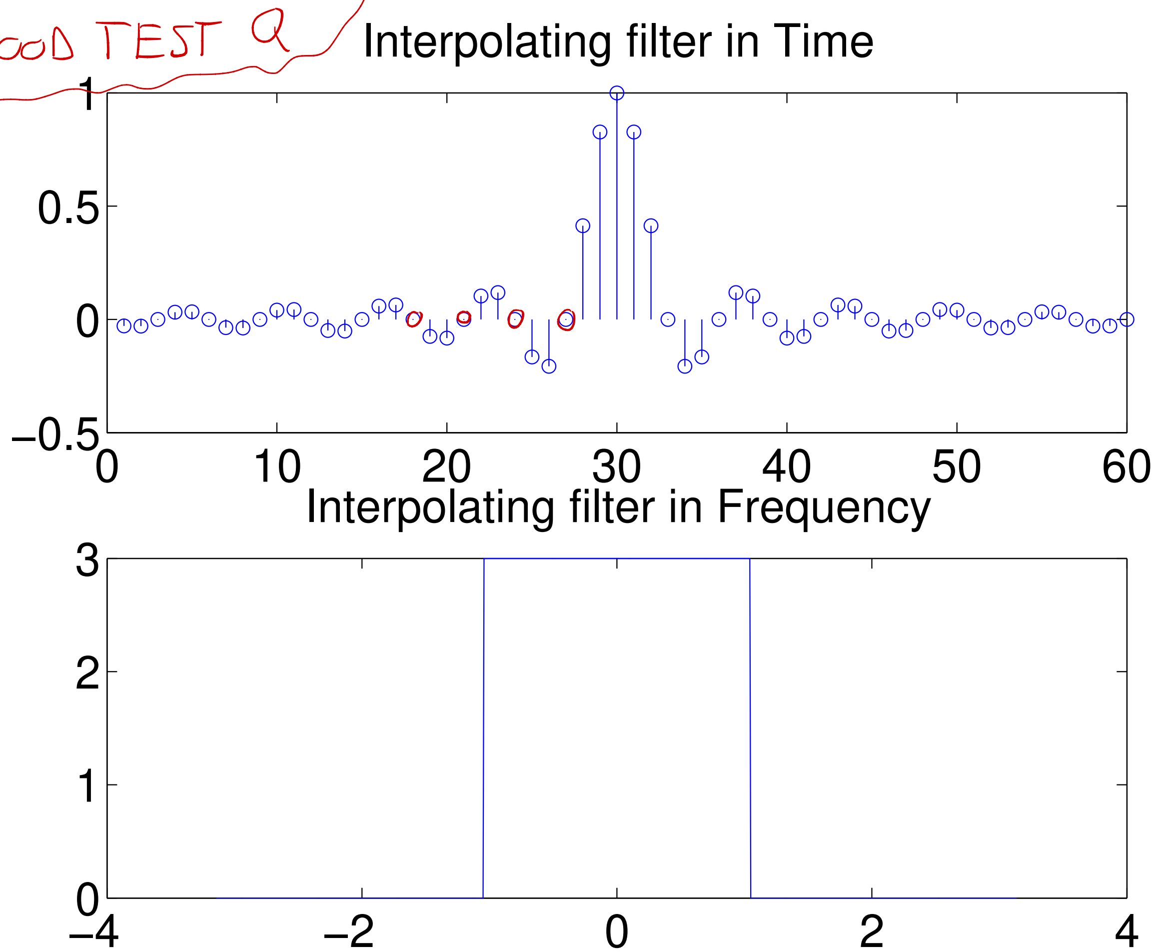
$$H(e^{j\omega}) = L\text{prect}_{2\pi/L}(\omega)$$

- In the time domain

convolve with this in time/space

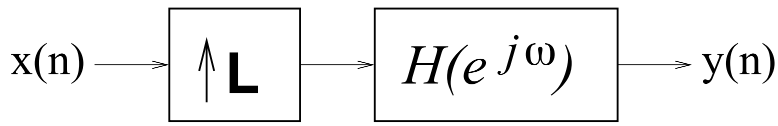
$$h(n) = \text{sinc}\left(\frac{n}{L}\right)$$

every 3rd sample here is zero
↳ GOOD TEST Q



injecting "redundant" but useful information

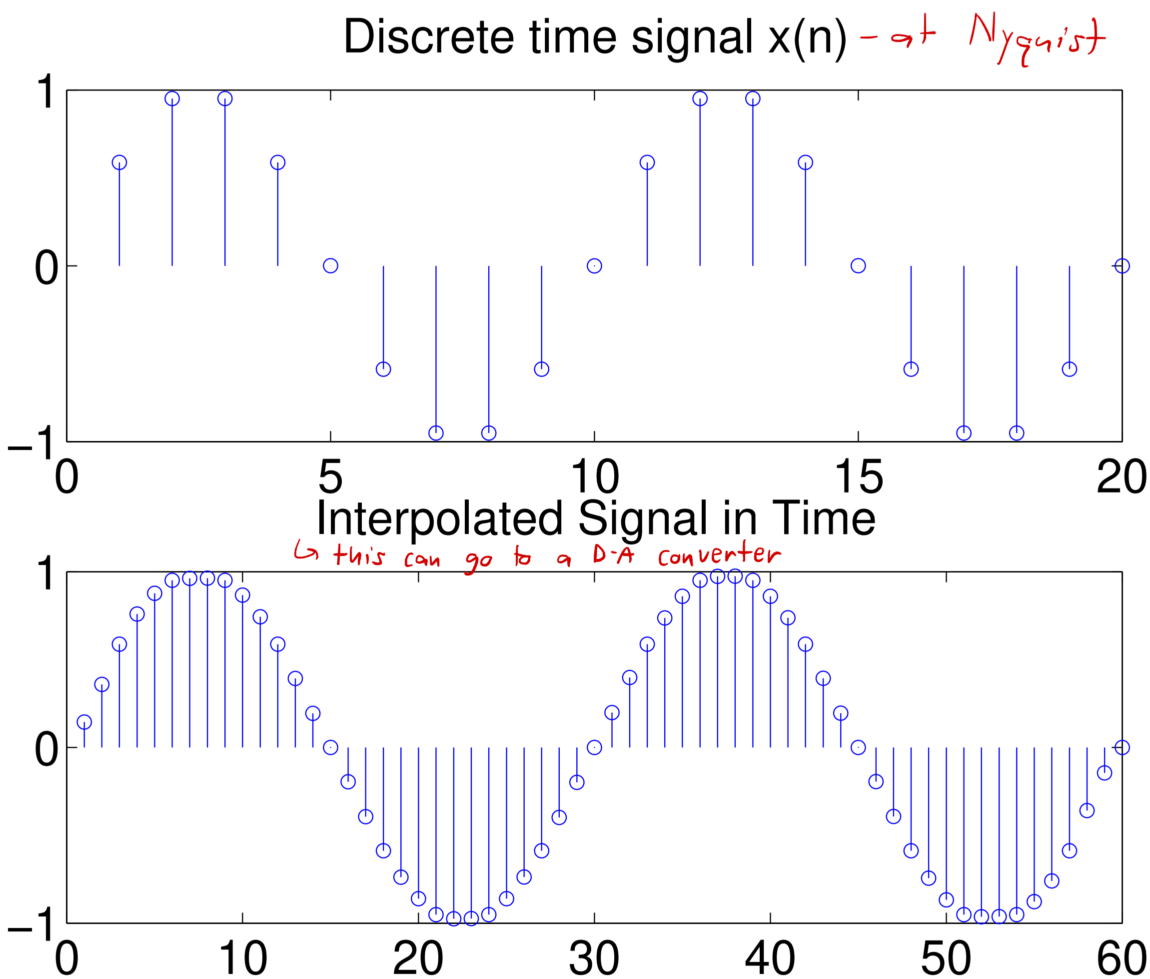
Interpolation in the Time Domain



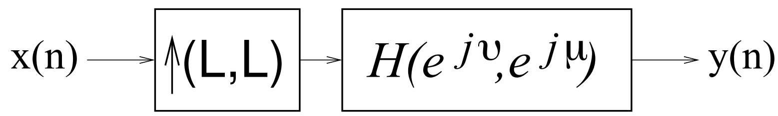
- In the frequency domain

$$Y(e^{j\omega}) = X(e^{j\omega L})$$

- Example for $L = 3$



2-D Interpolation



- Up-sampling

$$y(m,n) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x(k,l)\delta(m-kL, n-lL)$$

- In the frequency domain

$$H(e^{j\mu}, e^{j\nu}) = \text{prect}_{2\pi/L}(\mu)\text{prect}_{2\pi/L}(\nu)$$

$$h(m,n) = \text{sinc}\,(m/L, n/L)$$

- Problems: sinc function impulse response
  - Infinite support; infinite computation
  - Negative sidelobes; ringing artifacts at edges

Pixel Replication

↳ cheap & dirty way to approximate perfect sinc

$$x(n) \longrightarrow \boxed{\uparrow(L,L)} \longrightarrow \boxed{H(e^{j\nu},e^{j\mu})} \longrightarrow y(n)$$

- Impulse response of filter

$$h(m,n) = \begin{cases} 1 & \text{for } 0 \leq m \leq L-1 \text{ and } 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

- Replicates each pixel L^2 times

Bilinear Interpolation

$$x(n) \longrightarrow \boxed{\uparrow(L,L)} \longrightarrow \boxed{H(e^{j\nu}, e^{j\mu})} \longrightarrow y(n)$$

- Impulse response of filter

$$h(m, n) = \Lambda(m/L)\Lambda(n/L)$$

- Results in linear interpolation of intermediate pixels

↳ this creates a soft image, then would need to sharpen it

sample at f → max frequency in signal should be $\frac{f}{2}$ *

↳ Nyquist language